IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 11, November 2019

Adding Improvements to Multinomial Naive Bayes for Increasing the Accuracy of Aggressive Tweets Classification

Divisha Bisht[1], Sanjay Joshi[2]

Student, Department of Information Technology, College of Technology, GBPUA&T, Pantnagar, India[1]

Assistant Professor, Department of Information Technology, College of Technology, GBPUA&T, Pantnagar, India[2]

Abstract: Naïve Bayes is a popular supervised learning method widely used for text classification and sentiment analysis. There has been a rise of aggressive troll comments in the social networking sites which leads to online harassment and causes distressful online experiences. This paper uses Naïve Bayes classifier using Bag of Words on 'Tweets dataset for Detection of Cyber-Trolls' (dataset taken from Kaggle) and aims to improve baseline model by adding cumulative changes and studying their impact on the performance of the model.

Keywords: Naïve Bayes, Classification, Improvements, Accuracy

I. INTRODUCTION

The prevalence, ease of access and anonymity in the social media communities supported by the all dominating web of internet has led to a rise of online negative behaviour in the form of trolling. A regular influx of a large number of troll comments can be observed in most Social Networking Sites (SNS) like Twitter, Facebook, Reddit.

The Collins English Dictionary defines troll as "someone who posts unkind or offensive messages on social media sites, and often tries to start arguments with other users." [1] The comments posted by such users classified as troll comments are essential to identify such comments because they are harmful for the psychological state of both the intended subject of the troll as well as the other users who are caught in this malicious interaction.[2]

Aggression is used as a tool by online trolls to elicit various distressful psychological reactions from the people.[3] Aggressive remarks and hateful speech that may include use of profanities is a dominant characteristic of most troll comments. Therefore, detecting aggressive content becomes very important from the point of view of identifying troll comments.

The Naïve Bayes algorithm is one of the most popular and simple classification method under supervised machine learning techniques. Naïve Bayes classifiers are based on the Bayes theorem and assumes that the features or predictors are unrelated to each other. Using Bayes theorem, the posterior probability of a class given predictor can be calculated from prior probability of class, prior probability of predictor and likelihood (probability of predictor given class). The result of prediction by the classifier is the class with the highest posterior probability.

In this study, the Naïve Bayes algorithm is applied to 'Tweets dataset for Detection of Cyber-Trolls'[4] for classification of tweets into two classes- aggressive or not aggressive. First, a simple Bag Of Words (BOW) model is used for building the classifier. After that for further improvements techniques like Additive smoothing, removing stop-words and TF-IDF are applied to observe their effect on performance metrics of the classifier. This forms the basis if building a multinomial Naïve Bayes model with improved accuracy.

Rest of the paper describes Naïve Bayes model, proposed improvements methodology and the dataset used. Lastly, results are discussed followed by conclusions drawn from the study.

II. MATERIALS AND METHODS

A. Naïve Bayes classifier

Although Naïve Bayes is a probabilistic classifier which assumes the independence of features (hence 'naïve') , yet it provides extremely good results in comparison to other complex techniques.[5] It works on the Bayes theorem given by

DOI 10.17148/IJARCCE.2019.81108

Thomas Bayes (1701-1761) which finds the probability of an event, given the probability of an event which has already occurred :

$$\mathbf{P(C \mid x) = \frac{P(x \mid C) \, . \, P(C)}{P(x)}}$$

where,
P(C | x) - Posterior Probability of class C given feature x.
P(x | C) - Likelihood (probability of feature x given class C).
P(C) - Prior Probability of class C.
P(x) - Prior Probability of feature x.

The steps used by Naïve Bayes classifier are:

1. Find prior probability of the given class label.
2. Find the likelihood for each feature given the class label.
3. Calculate the posterior probability from the above formula.
4. The class label with the highest posterior probability is the result of the prediction.

After appropriate vectorization of the dataset using Bag of Words technique. MultinomialNB classifier of the Scikit-Learn library is imported and fit to the training set,

B. Bag of Words Model

Since Naïve Bayes works on numerical data, each document is converted into a feature vector using Bag of Words. First a vocabulary of all the words in the entire corpus is built. This vocabulary consists of all the distinct words present in the corpus along with the word count. If the vocabulary is n- dimensional, then for each document in the text, an n- dimensional feature vector is created. Hence, the Bag of words is based on the assumption that the position of words in the document does not matter.[6]
The CountVectorizer function of the Scikit- learn library is used for implementing the Bag of Words. This function performs tokenization, counting and normalization to extract the numerical feature from the text.[7]

C. Improvements added to the classifier

1) Additive Smoothing: To solve the problem of zero probabilities, additive smoothening has been applied to the MultinomialNB classifier by choosing the value of the smoothing parameter α as 0.001 after running the model on different values and observing the results. This case where $\alpha>1$ is known as Lidstone smoothing.

2) Stop- words: Stop words are the extremely common words in the vocabulary of the corpus and can be removed from the vocabulary using a collection of stop words called stop list.[8]
In this study, the stop_words parameter of the CountVectorizer function of Scikit- learn library is used for removal of stop words from the tokens using the built- in 'english' stop list.

3) TF-IDF feature: The term frequency - inverse document frequency (TF-IDF) is the product of the Term Frequency (TF) and Inverse Document Frequency (IDF). Term Frequency is the number of times a term occurs in the document while Inverse document frequency is the total number of documents in the corpus divided by the number of documents containing the term and taking the log of the result.[9] Less frequent words have high TF-IDF score whereas low- TF-IDF scores are assigned to commonly occurring words in all the documents. For incorporating TF-IDF feature, the TfidfTransformer function of the Scikit- Learn library is used the Bag of Words model.

D. D. Proposed Method

The proposed method is depicted making use of Fig.1 shown below in which first, a baseline Naïve Bayes model is built using the Bag of Words Technique. Over this, additive smoothing is applied to resolve the problem of zero probabilities.

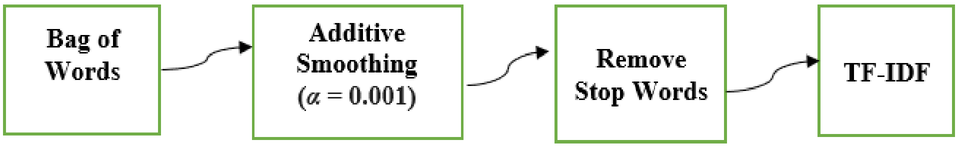

Fig. 1 Successively adding improvements to the baseline Model

After that, stop words are removed making use of a stop- list. All this is followed by adding TF-IDF feature to the existing Bag of Words model to observe the effect on performance metric. In this paper, effect of applying the above techniques on top of baseline model of Naïve Bayes is observed. After every successive optimization the result is compared to determine the best optimization sequence which gives reasonably good accuracy over baseline model for the given dataset.

E. Dataset Description

'Tweets dataset for Detection of Cyber-Trolls'[4] is the dataset taken from Kaggle which is used in this study. The dataset is human labelled consisting of 20001 tweets in a json file format. The dataset has all the items manually labelled into following 2 categories[10]:

1 (Cyber-Aggressive)
0 (Non Cyber-Aggressive)

Pre-processing of the dataset is performed using python to convert the dataset into a corpus that can be used for the Bag of Words model. The libraries used in this study are numpy, pandas and scikit- learn.

III. RESULTS AND DISCUSSION

With each optimization added, the performance of the Naïve Bayes classifier is observed. The effect of every optimization added on the accuracy of classifier is observed and a final improvement model with a reasonably good accuracy is defined.

The IDE used is Spyder 3.2.8 and the python libraries used are pandas, numpy and scikit-learn. The performance metrics used are accuracy, precision, recall and F1 score.

The performance of the classifier after each cumulative step of improvement is shown in Table I. The same has been depicted in graphical form in Fig. 2.

Table I Experimental Results

Improvements (added cumulatively)	Accuracy	Precision	Recall	F1 Score
Bag of Words	0.7844	0.7301	0.7115	0.7207
Additive Smoothing (α = 0.001)	0.8050	0.7264	0.8040	0.7632
Remove stop words	0.8056	0.7299	0.7979	0.7624
TF-IDF	0.8212	0.7566	0.8000	0.7777

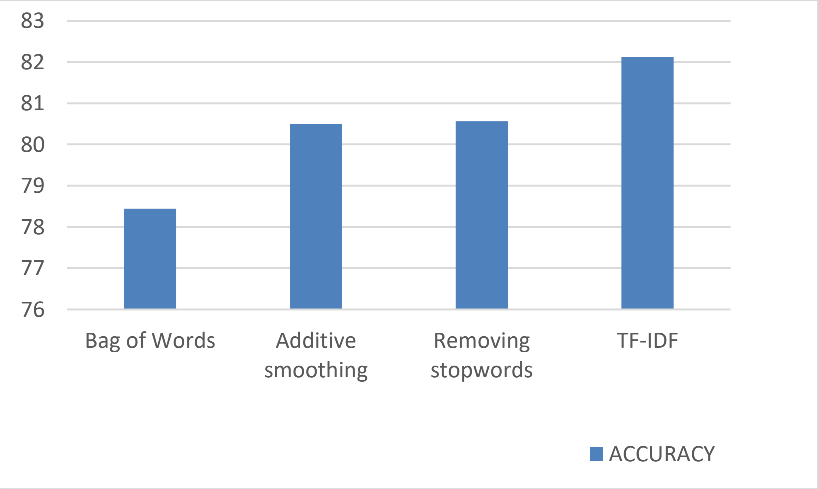

Fig. 2 Accuracy of the model for Each Improvement Added

IV. CONCLUSION

The above results conclude that some improvements like removal of stop words do not have a significant impact on the accuracy of the model while improvements like additive smoothing and TF-IDF give improved accuracy. A substantial improvement in the model with additive smoothing ($\alpha = 0.001$) is observed as it handles the zero probabilities problem. Removing stop words caused a negligible improvement in the baseline probably due to most tweet lengths being small and therefore, excluding common words did not make a significant difference. Results also point that by adding TF-IDF scores on top of baseline model which has already been smoothed give us a reasonably good accuracy of 82.12%. The improvements done to the baseline model are simple yet effective and efficient. For future considerations, other advanced optimizations can be explored and added to the baseline model for increased accuracy.

REFERENCES

[1]. Definition of ‘troll’, Collins English Dictionary. [Online]. https://www.collinsdictionary.com/dictionary/english/troll
[2]. L.G Mojica., “Modeling trolling in social media conversations”, arXiv:1612.05310 [cs.CL], 2016.
[3]. Internet trolls and their Aggression. [Online]. Available: https://exploringyourmind.com/internet-trolls-aggression/
[4]. Tweets dataset for Detection of Cyber-Trolls. [Online]. Available: https://www.kaggle.com/dataturks/dataset-for-detection-of-cybertrolls/activity
[5]. Irina Rish, “An empirical study of the naive Bayes classifier”, IJCAI 2001 workshop on empirical methods in artificial intelligence, Vol. 3, No. 22, 2001.
[6]. Sebastian Raschka, “Naive bayes and text classification i-introduction and theory”, arXiv preprint arXiv:1410.5329, 2014.
[7]. Feature extraction. [Online]. Available: https://scikit-learn.org/stable/modules/feature_extraction.html
[8]. Dropping common terms: stop words. Online: https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html
[9]. TF-IDF. [Online]. Available: https://en.wikipedia.org/wiki/Tf%E2%80%93idf
[10]. A. Narayan. [Online]. Available: https://dataturks.com/projects/abhishek.narayanan/Dataset%20for%20Detection%20of%20Cyber-Trolls

DOI 10.17148/IJARCCE.2019.81108